Pragmatic Interpretation and Ambiguity

Charles E. Martin
Yale University

Abstract

An approach to pragmatic interpretation in natural language understanding is described. The approach trades off a full generative natural language capacity for the ability to recognize the flow of familiar (and often complex) arguments.

The theory requires considerable domain-dependent knowledge and specific domain-dependent goals for the understanding system. The process model described, however, is domain-independent with fairly relaxed representational constraints. All processing takes place within a hierarchical episodic memory, allowing expectations to be posted to quite general concepts from multiple sources in parallel.

INTRODUCTION

The Direct Memory Access Parser (Riesbeck and Martin, 1985, Martin, 1989) uses phrases to guide memory search. DMAP tries to recognize specific concepts in memory; new memory structures are added to reflect differences between these concepts and the input.

The DMAP system is an example of a *pragmatic* natural language understander; it understands new texts only in terms of existing memory structures. Interpretation is determined by the character of these existing concepts: at any time, its interpretive goals—and thus the range of target concepts for understanding—are determined by which existing concepts are in the process of being recognized.

The simultaneous strength and weakness of DMAP is its reliance on pre-existing memory structures. The phrases used in parsing are tied to existing concepts; the sequence of elements that make up the phrase constitute the only store of linguistic knowledge in the system. This resembles the familiar "pattern-concept" pairs of phrase-based systems (Arens, 1981). The key difference between past phrasal systems and DMAP is that the *concept* is used to retrieve the *pattern*, rather than the reverse.

Instead of using a pattern to *build* a concept, a predicted concept is used to retrieve patterns which will suffice to *recognize* that concept. The system has expectations about the content of the text in advance of seeing the text itself. The patterns serve as a mechanism to *verify* predictions made by the system. As in traditional phrase-based systems, some pattern elements appearing in the text will differ from their general specification in the pattern, and new concepts are built to reflect these differences.

INTERPRETATION AND AMBIGUITY

At the most general level, natural language understanding systems for semantic interpretation can be thought of as choosing which of several hypotheses best explains the input text. For example, a traditional conceptual analyzer (Riesbeck, 1974) might retrieve a lexically-indexed production containing information to determine which of several competing knowledge structures should be selected as the proper representation for a word. Modern marker-passing schemes such as the WIMP parsers of Charniak (1986) or Norvig's FAUSTUS (1987) take a conceptually similar approach: a marker-passing or spreading activation algorithm identifies candidate hypotheses in the form of explanatory memory structures; the choice between candidates is determined by an evaluation metric of some kind.

The high-level structuring of semantic interpretation into the component parts (1) find candidates and (2) select among them puts a tremendous amount of responsibility on the evaluation metric. In the case of a conceptual analyzer, this is reflected in increasingly larger **cond** clauses (or mutually antagonistic demons) which seem ad hoc, disassociated from memory, and hard to learn. For marker-passers and their brethren, this is reflected in large numbers of false candidates and a "hit rate" of valid inference which decreases with the size of the memory. These problems are well known.

The general theory behind DMAP is that interpretation is easier given a different breakdown of the problem. First, the understanding system is assumed to have some *expectations* about what the input text will be about. Second, given these expectations, the interpretation problem is cast as *verifying* that these expectations are fulfilled. Hypothesis verification is generally easier than determining hypothesis applicability. This breakdown is the basis for script and frame-based parsers such as SAM (Cullingford, 1978), later extended to include much more of the inferential processes of memory in (Schank, Lebowitz, and Birnbaum, 1980).

This traditional theory of scriptal expectations has well-known limitations. It is unclear what is to occur when more than one script is active at a time; if each has its own lexicon, the problem of choice among alternatives has simply reappeared under a different guise. Second, more than one word sense may match active expectations; at this point the script-based analyzer must simply choose one. See (Birnbaum, 1986) for a fuller critique.

DIRECT MEMORY ACCESS PARSING

The solution advocated to these problems is to recast the theory of scriptal expectations in terms of predicting memory structures and relationships in memory, rather than isolated representational structures. Using a hierarchical episodic memory based on the Memory Organization Packets of (Schank, 1982), the system allows any concept in memory, at any level of generalization, to post expectations to another concept or lexical item. The general DMAP algorithm breaks this process of posting expectations and verifying predictions into two components: directed prediction, and opportunistic recognition.

Directed prediction

1. Predict a concept.
2. Retrieve associated patterns which will recognize that concept.
3. Predict patterns' leading edges (can be concepts or lexical items).

Opportunistic recognition

1. Activate a concept.
2. Find all referenced predictions.
3. Refine to more specific predictions.
4. If the prediction's pattern is complete *then* activate the predicted concept *else* predict the pattern's next element.

The system reads articles about the economy; initial predictions come from high-level memory structures such as "article about interest rates." Specific predicted concepts may use patterns associated with more abstract concepts; as input is recognized, the activated concepts may provide information allowing the predicted concepts to be refined to even more specific concepts in memory. This use of specific memory structures as interpretations for new input is similar to recent techniques of case-based reasoning (see Kolodner, 1988, for a variety approaches).

Patterns are the sole store of pragmatic knowledge in the system. When a target concept is predicted, the patterns supply expectations about what other concepts the system should *attempt* to recognize in order to recognize the target. There are two restrictions on patterns:

1. *(Index patterns)* How expectations are posted must be represented as a linear sequence of lexical items and references to other concepts, and
2. *(Relative reference)* References to other concepts must be expressed in terms of the packaging relationship between the concepts and the source of the expectation.

As a trivial example, the concept MTRANS for communication events is a relatively general concept in its hierarchy, which includes specific instances of communication below it and more general concepts of action above it. The MTRANS concept itself packages (in the usual labelled part-subpart relationship of semantic networks) concepts for the *(actor)* of the action and the *(content)* of the communication. A simple index pattern for this concept would be:

{ *(actor)* **says** *(content)* } ⟶ MTRANS

If MTRANS is predicted, then the index pattern can be applied. To apply the index pattern, the first element of that pattern is examined. If it is a lexical item, then the lexicon is updated with the information that that lexical item refers to that index pattern. If it is a packaging relationship, the concept packaged serves to index the expectation *directly in the memory structure referred to by that packaging relationship.* This latter memory structure is "predicted."

The index patterns provide a distributed representation of pragmatic lexical and conceptual knowledge in the system, in the sense that the same elements of a pattern (lexical or referential) may appear in different patterns attached to quite different concepts in memory. For example, "come back to" appears in the phrase { `come back to` *(topic)* } associated with the concept INTERVIEW-TOPIC-INTRODUCTION, and also in the phrase { *(actor)* `come back to` *(plan)* } associated with the concept REPEATED-PLAN. Since there is no central generative definition of "come back to," the priming of these phrases is dependent only upon the expectations in memory. The first concept (and pattern) will be predicted while reading a newspaper interview, while the second will be predicted in explaining the actions of economic actors.

It is arguable that this pragmatic, concept-specific, distributed representation of patterns loses some of the generative capacity of our linguistic knowledge; surely there is some deeply-rooted similarity between these two phrases and their concepts. But the direct indexing of expectations in memory structures is crucial to allowing expectations to be placed on any concept in memory, no matter how specific or general. The existence of an abstraction hierarchy is then exploited by indexing the expectation under all abstractions of the predicted concept. This is "marker passing," but only according to the rigid constraints of the hierarchy and the predicted concept.

Marker Passing[1]

When concepts are recognized, they spread their recognition up the abstraction hierarchy as well. Intersections of expectation and recognition causes further predictions based on the index patterns associated with the expectation. When an index pattern is completed, it results in the recognition of its predicted concept.

Intersections also result in the refinement of the system's predictions. Given a prediction for MTRANS, knowledge that the actor of the communication is Milton Friedman allows the prediction of communication to be refined to specific episodes in memory in which Milton Friedman has had something to say. Once at this specific level, information unique to understanding the kinds of arguments Milton Friedman makes (for example, monetarist arguments) is available to the system. The memory search algorithm therefore acts as a kind of giant script applier, in which concept refinement can cause scriptal expectations to be posted at any level of abstraction or packaging.

Intersections occur in DMAP only as a result of the guided passing of markers, not as a result of blind memory search. Since the index patterns which guide this passing of markers are tied to domain-dependent memory structures, the algorithm as a whole behaves as determined by domain-dependent knowledge even though the algorithm itself is domain-independent.

Prediction Failure

The DMAP system uses the input to supply specific information to confirm or deny the predictions in memory. The refinement of those predictions supplies additional, specific expectations based on the concepts already existing in memory: "Ah, it's Friedman's column in *Newsweek*. He must be talking about the money supply." For the most part, these expectations are satisfied, and the system does quite well with many vague or ambiguous referents: "He's talking about an 'economic nightmare.' He must be referring to the increase in the money supply."

Quite often, however, these expectations are too specific, or just incorrect. For example, Friedman might call for the active application of expansionary money supply—not expected, but possible. Such an input creates an *expectation failure* for DMAP, in which the input did not match the prediction.

Most interesting texts contain expectation failures. DMAP treats them as specific input used to recognize memory structures representing the failure and a possible repair. These failure and repair structures are represented in the same memory format as all other concepts, and operated on by the same *directed prediction* and *opportunistic recognition* algorithms. The difference is that an internal processing distinction—the mismatch of input and prediction—is used as input to the recognition process, instead of an external text. Space prohibits discussing the failure and repair mechanisms further, but see Martin and Riesbeck, 1986, and Martin, 1989 for details.

UNDERSTANDING ECONOMIC ARGUMENTS

This section presents a detailed example of how DMAP runs on the following input text.

[1]Although the theory makes no committment to marker passing, it is a convenient device for explanation. The program is implemented as a marker passer. See (Martin, 1989) for details.

Donald Regan: On a long term basis, interest rates are headed down.

This is part of a very long text handled by the parser. The parser has the problem of connecting the parts of the text to memory. It's domain goals are to determine an explanation for possible future states of the economy; in other words, the program is interested in figuring out what will happen to interest rates. Represented in memory is specific domain knowledge as well as general knowledge about argumentation (Birnbaum, Flowers, and McGuire, 1980).

The example presents successive words as they are seen by the parser. Following each word are the index patterns whose expectations have been satisfied by the reference to the input word. With the index pattern is the associated concept in memory. An asterisk (*) indicates what remains of the pattern to be recognized. When a pattern is complete, the concept is noted as `referenced`, and expectations which it satisfies will follow with their index patterns and associated concepts follow.

Program annotations include:

- `Specialization Failure`, which reports which relationship failed and what new structure was built or used.
- `Refinement Failure`, which reports what new structure was built or used, the source of the failure, and any repairs which were performed.
- `Repair`, which reports new memory structures built to record the repair and what the content of those concepts are.

```
Reading DONALD
    { DONALD * REGAN } = REGAN
Reading REGAN
    { DONALD REGAN * } = REGAN referenced
    { (actor) * (mtrans) (mobject) } = IR-UP-COMMUNICATION
        Specialization Failure: (actor) REGAN
        Built: IR-UP-COMMUNICATION.1
```

This demonstrates the parsing of a simple lexical phrase. Recognition of the index pattern results in the reference of the `REGAN` concept, which in turn satisfies an expectation from `IR-COMMUNICATION`. There is no refinement specific to Regan, so a new specialization is built.

```
Reading :
    { : * } = MTRANS referenced
    { (actor) (mtrans) * (mobject) } = IR-UP-COMMUNICATION.1
Reading ON
    { ON * A (horizon) BASIS } = BEHAVIOR
Reading A
    { ON A * (horizon) BASIS } = BEHAVIOR
Reading LONG
    { LONG * TERM } = LONG-RUN
Reading TERM
    { LONG TERM * } = LONG-RUN referenced
    { ON A (horizon) * BASIS } = BEHAVIOR = TRUE-BEHAVIOR
```

This is the normal refinement process; the associated concept for this index pattern was BEHAVIOR, but the reference in the input to a LONG-RUN (horizon) caused refinement of the concept to that of TRUE-BEHAVIOR. (The underlying model of economic reasoning being that the long-term action of an economic quantity reflects its true behavior, rather than its (uncertain) short-term activity.)

It is worth noting here that the concept for "long" was expected because the higher-level index pattern for BEHAVIOR was looking for its horizon relationship. "Long term" was connected to this concept because of the presence of the higher-level expectation.

```
Reading BASIS
    { ON A (horizon) BASIS * } = TRUE-BEHAVIOR referenced
    { (behavior) * (event) } = TRUE-TREND
    { (behavior) * (quantity) (action) } = IR-INCREASE
        Specialization Failure: (behavior) TRUE-BEHAVIOR
        Built: IR-INCREASE.2
Reading RATES
    { RATES * } = RATES referenced
    { (topic) (mention) * } = INTEREST-RATES referenced
    { (behavior) (quantity) * (action) } = IR-INCREASE.2
```

"Interest rates" is the topic of the entire sequence of newspaper texts, and the parser has already represented this from its prior processing. Interest rates are often referred to simply as "rates," and the second index pattern in the immediately preceeding trace captures the connection between the topic concept and its corresponding mention concept. Recognition of this sequence from the mention results in the topic concept being referenced.

```
Reading ARE
    { ARE * HEADED (direction) } = INCREASE
Reading HEADED
    { ARE HEADED * (direction) } = INCREASE
Reading DOWN
    { DOWN * } = DOWN referenced
    { ARE HEADED * (direction) } = INCREASE
        Refinement Failure: (direction) is DOWN not UP
        Using: DECREASE
        Source: IR-INCREASE.2 no repair
```

This is the first refinement failure so far. The expectations from the previous texts have generated an expectation at this level for another argument supporting a future increase in interest rates. The text does not support this interpretation, and so the (already constructed) text-supported interpretation DECREASE is referenced. The parser attempts to *repair* the failure by recovering the source of the expectation. Unfortunately, no repair structures are able to be found from this failed refinement.

```
    { (behavior) (quantity) (action) * } = IR-INCREASE.2
        Refinement Failure: (action) is DECREASE not INCREASE
        Built: IR-DECREASE.3
        Source: IR-UP-COMMUNICATION.1 no repair
```

This index pattern recognizes the argument for a particular economic event. Once again, the prior expectations have failed, but no repair is possible.

It should be stressed that the lack of repairs is a function of the knowledge of the system; in other words, there is no *a priori* reason why sensible repairs for this failure might not be present.

```
{ (behavior) (event) * } = TRUE-TREND
    Specialization Failure: (event) IR-DECREASE.3
    Built: TRUE-TREND.4
```

The TRUE-TREND structure essentially represents the concept mentioned in passing in the annotations following the word "long," above. Regan's argument results in the recognition of this index pattern and the reference of this concept. There is no refinement failure here, but a new specialization must be built.

```
{ (actor) (mtrans) (mobject) * } = IR-UP-COMMUNICATION.1
    Refinement Failure: (mobject) is IR-DECREASE.3 not IR-INCREASE.2
    Built: IR-COMMUNICATION.5
    Source: SUPPORT-IR-UP (argument) found REPAIR:FAILED-SUPPORT
    Repair: ATTACK-POINT
    Built: ATTACK-POINT.6
        (point) IR-INCREASE
        (basis) TRUE-TREND.4
    Built: SUPPORT.8
        (argument) IR-COMMUNICATION.5
    Built: REPAIR.7
        (source) SUPPORT.8
        (expected) IR-INCREASE
        (reference) IR-DECREASE.3
        (repair) ATTACK-POINT.6
```

The failure of the argument communicated by Regan to match the expectations from prior texts finds a suitable repair. This repair, REPAIR:FAILED-SUPPORT, represents knowledge about how to argue: to challenge a point, you may attack it directly. In this case, since the parser is trying to come to a conclusion about the behavior of interest rates, challenging the previous texts' main point may be done by asserting that interest rates will decrease.

The repair is to modify memory to include a new attack structure, which contains packaging relationships to the point attacked and the basis for the attack. Since the TRUE-TREND structure is now part of memory, it is seized upon as a reasonable part of the attack structure. In paraphrase, this new concept represents "interest rates will not increase because decreasing interest rates reflect the true long-term behavior of the quantity."

```
{ (support) (attack) * } = DISPUTED-IR
    Specialization Failure: (attack) ATTACK-POINT.6
    Built: DISPUTED-IR.9
```

The fact that previous texts supported the opinion that interest rates would rise means that the current attack on that opinion indicates that the opinion is disputed. A new specialization must be built to record this interpretation.

Finally, the fact that the system had already come to an internal conclusion about interest rates based on its previous understanding provokes an internal response to the attack. Although the notion that there is concord about the behavior of interest rates is not necessarily the final decision of the system, it prompts index patterns which can be used to discredit the attack to generate expectations looking for some basis for an attack on the attack.

CONCLUSIONS

The point of the DMAP approach is to make interpretation wholly dependent on the expectations in memory. DMAP presents a theory of understanding based on script-frame theory, but updated to our modern concepts of hierarchical memory organization. As implemented, it works on quite large examples in the absence of any specialized domain-dependent control structure; all knowledge is represented in a typical abstraction and packaging hierarchy and linear sequences of index patterns.

Acknowledgements. Thanks to anonymous reviewers for their comments. This work was supported in part by the Defense Advanced Research Projects Agency, monitored by the Office of Naval Research under contract N0014-85-K-0108 and by the Air Force Office of Scientific Research under contract F49620-88-C-0058.

REFERENCES

Arens, Y. (1981) Using Language and Context in the Analysis of Text. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, B.C., Canada.

Birnbaum, L. (1986) *Integrated Processing for Planning and Understanding.* Ph. D. Thesis, Yale University.

Birnbaum, L., Flowers, M., McGuire, R. (1980) Towards an AI model of argumentation. *Proceedings of AAAI-80.*

Charniak, E. (1986) *A neat theory of marker passing. Proceedings of AAAI-86.*

Cullingford, R. (1978) *Script application.* Ph. D. Thesis, Yale University.

Kolodner, J.L. (ed.) (1988) *Proceedings: Case-Based Reasoning Workshop.* Morgan Kaufman.

Martin, C.E., and Riesbeck, C.K. (1986) *Uniform Parsing and Inferencing for Learning. Proceedings of AAAI-86.*

Martin, C.E. (1989) *Direct Memory Access Parsing.* Ph. D. Thesis, Yale University. Forthcoming.

Norvig, P. (1987) *Inference in text understanding. Proceedings of AAAI-87.*

Riesbeck, C.K. (1974) *Conputational Understanding: Analysis of Sentences and Context.* PhD thesis, Stanford University, Stanford, CA.

Riesbeck, C.K., and Martin, C.E. (1985) *Direct Memory Access Parsing.* YALEU/DCS/RR 354, Yale University.

Schank, R.C., Lebowitz, M., and Birnbaum, L. (1980) An integrated understander. *Am. J. Comp. Ling.* **6**.

Schank, R.C. (1982) *Dynamic Memory: A Theory of Learning in Computers and People.* Cambridge University Press.